

2013 State of DevOps Report

PRESENTED BY
puppet
labs &
IT REVOLUTION PRESS

Key Findings

In December of 2012, Puppet Labs and IT Revolution Press surveyed over 4,000 IT Operations professionals and developers in the largest industry survey of this magnitude. The survey results revealed accelerating adoption of DevOps practices in IT organizations across companies of all sizes. Sixty-three percent of respondents have implemented DevOps practices, a 26 percent increase in adoption rate since 2011.¹

Respondents from organizations that implemented DevOps reported improved software deployment quality and more frequent software releases. Our findings substantiate the business value of DevOps, too: It enables high performance by increasing agility and reliability. We found that high performing organizations:

Ship code 30x faster

and complete those deployments 8,000 times faster than their peers.

Have 50% fewer failures

and restore service 12 times faster than their peers.

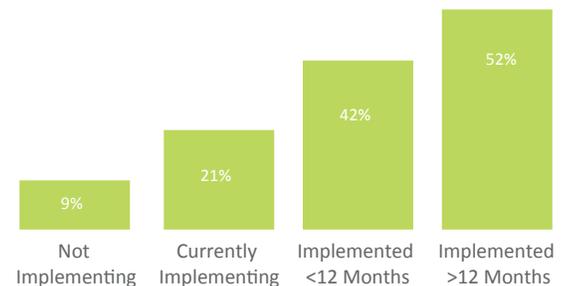
Not surprisingly, organizations that have implemented DevOps practices are up to five times more likely to be high-performing than those that have not. In fact, the longer organizations have been using DevOps practices, the better their performance: The best are getting better.

DevOps adoption is at a tipping point. Adopting DevOps practices improves performance, and the longer those practices are in place, the greater the performance advantage. Waiting to implement is no longer an option. Following our recommendations will put you ahead of the curve as DevOps approaches critical mass.

ORGANIZATIONS THAT HAVE IMPLEMENTED DEVOPS SAW THESE BENEFITS:



HIGH PERFORMANCE BY DEVOPS MATURITY



¹ Respondents of both the 2011 and 2012 survey are likely to be early adopters, making the increase in DevOps adoption rate much more telling of the overall growth of DevOps during this time period.

Who Took The Survey?

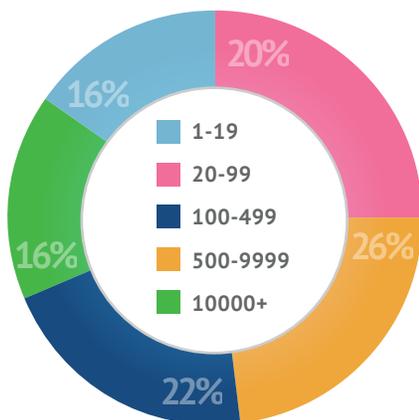
We saw respondents from over 90 countries across organizations of all sizes. The majority of respondents identified themselves as administrators, engineers, or developers, with only 16 percent manager-level or above.

Respondents worked for organizations running the gamut from startups to small/medium enterprises to WebOps giants, revealing that DevOps is happening everywhere.

SURVEY AREA



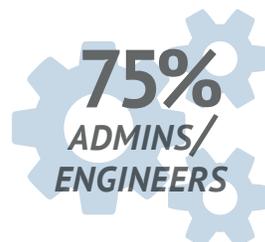
COMPANY SIZE (# OF EMPLOYEES)



DEPARTMENTS



ROLES



DevOps Increases Agility and Reliability

IT organizations are expected to respond more quickly to urgent business needs while simultaneously providing stable, secure, and predictable IT service. However, the systems on which the business operates are typically fragile and hostile to change. Adopting Agile development processes without improving operational reliability or communication between developers and operations only makes this problem worse. The increased frequency of releases from development creates even more of a burden on an already strained IT organization. Similarly, adopting rigid ITIL/ITSM standards without addressing development issues and improving communication channels results in an inflexible IT organization that simply cannot respond to business needs quickly enough. DevOps picks up where Agile methodology and IT standards left off.

For the first time ever, by adopting DevOps practices, IT organizations can be simultaneously agile and reliable, thus fulfilling the promise of over 20 years of Agile and ITIL/ITSM.

High performing organizations:

Deploy code 30 times more frequently.

High performing organizations deploy code 30 times more often, and 8000 times faster than their peers, deploying multiple times a day, versus an average of once a month. Frequent deployments coupled with faster change lead times enable operational agility.

Have 50 percent fewer failures.

High performing organizations have double the change success rate and restore service 12 times faster than their peers. Fewer failures and faster recovery mean less risk to the business when changes are deployed.

DevOps practices enable IT organizations to quickly and safely deploy changes, freeing them to work on higher level business objectives. In the next section, we break out four key metrics that show the impact DevOps has on performance.



Rob Cornish, CTO at the International Securities Exchange (ISE), a leading US options exchange that exceeds daily trading volumes of 2.5 million contracts.

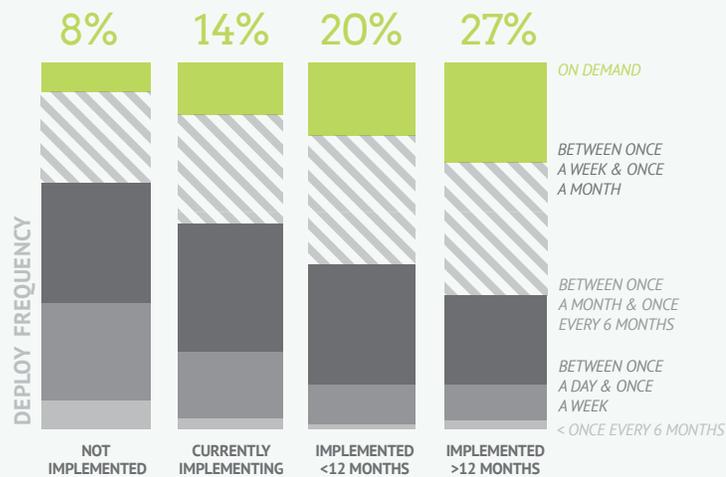
"DevOps has been instrumental in reducing our software deployment cycle from days to minutes. We are able to deploy new features to our customers quickly and safely, and rapidly respond to market changes."

How DevOps Enables High Performance

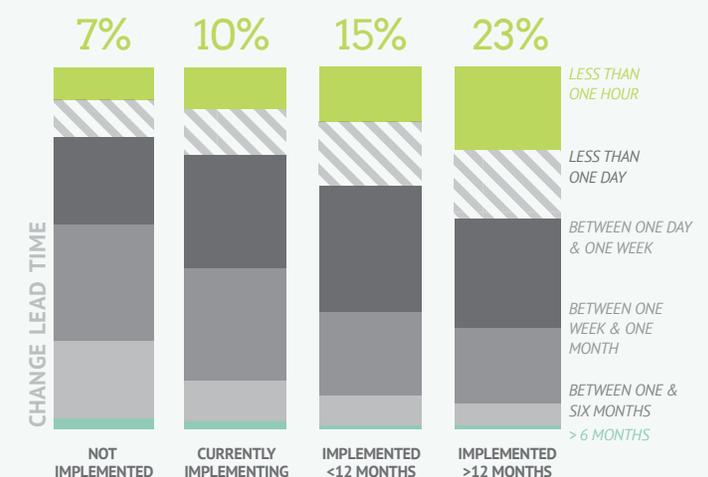
A high-performing organization is characterized by its ability to ship business-critical applications quickly without disrupting service. The classic examples of this are big WebOps shops—Google, Amazon, Twitter, and Etsy—known for deploying multiple times a day.

We found that organizations that had implemented DevOps practices were up to five times more likely to be high-performing than those that had not, and the more mature the implementation, the higher the performance.

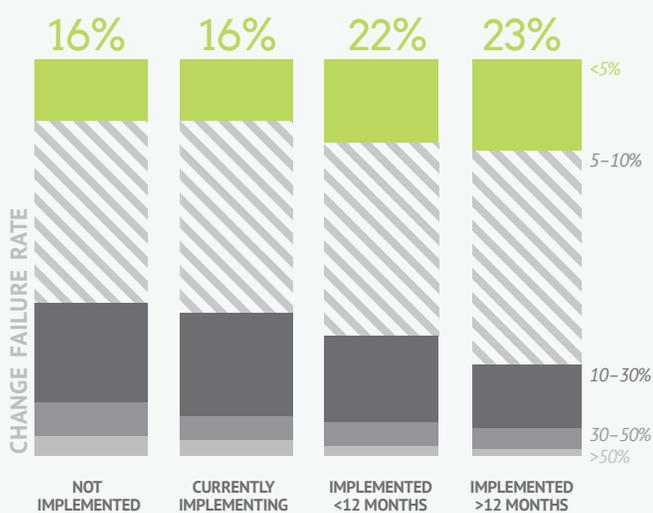
DEPLOY FREQUENCY



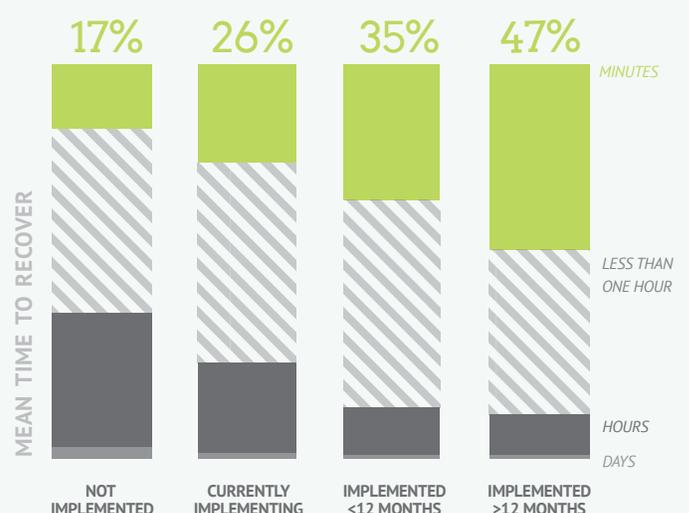
CHANGE LEAD TIME



CHANGE FAILURE RATE



MEAN TIME TO RECOVER



We analyzed four key DevOps performance metrics—deploy frequency, change lead time, change failure rate, and mean time to recover—by DevOps maturity, ranging from not implemented to implemented over 12 months ago. Organizations with more mature DevOps implementations saw significantly higher performance across all metrics compared to those that had not yet implemented DevOps.

Key Agility Performance Indicators

DEPLOYMENT FREQUENCY

High-performing organizations deploy at least once a week, and often multiple times a day. On average, this is 95 percent less time between deployments than lower performing organizations, enabling these organizations to quickly respond to market changes or customer feedback, and iterate on new ideas.

CHANGE LEAD TIME

Being able to quickly make changes is a basic measure of agility. If your infrastructure is bogged down by technical debt, the time to get code successfully running in production is significantly longer and your changes are more likely to fail. High-performing organizations make changes with a few minutes' notice, while their peers have change lead times of up to a month. Agile organizations can make 8,000 changes before their slower competitors can vet and deploy a single change.

Key Reliability Performance Indicators

MEAN TIME TO RECOVER

High-performing organizations recover from outages within minutes, compared to lower performing organizations that take an hour or longer to recover: over 30 times faster. Most organizations can't afford even a few minutes of downtime—that can mean thousands of lost transactions, resulting in lost revenue and lower customer satisfaction.

CHANGE FAIL RATE

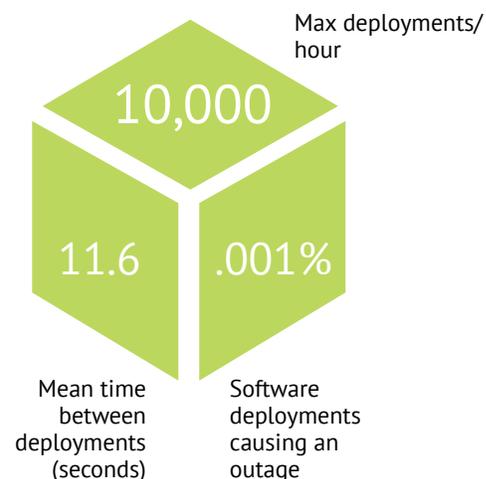
When changes fail, your organization has to devote time and resources to recovering, usually by taking those resources from developing, releasing, and supporting new products or updates. High-performing organizations have over 50 percent fewer failures from code changes. Changes often fail because the development and test environments don't match the production environment.

Obamas DevOps Success



Using DevOps practices, Obama's tech team was able to manage 2,000 nodes across three data centers, processing over 180TB and 8.5 billion requests, with just 30 minutes of downtime during the entire 18 months of the election campaign.²

DEVOPS VALUE IN ACTION: VELOCITY AT AMAZON AWS³



²Condliffe, Jamie, "Inside Obama's Tech Team." <http://gizmodo.com/5960790/inside-obamas-team-tech>. Nov 15, 2012.

³Jenkins, Jon. "Velocity Culture." Keynote, Velocity Conference, Santa Clara, Jun 16, 2011.

How to Achieve High Performance

Now that we know what high performance looks like, how do you actually achieve it? High-performing organizations share two common practices:

- 89 percent use version control systems for infrastructure management
- 82 percent have automated their code deployments

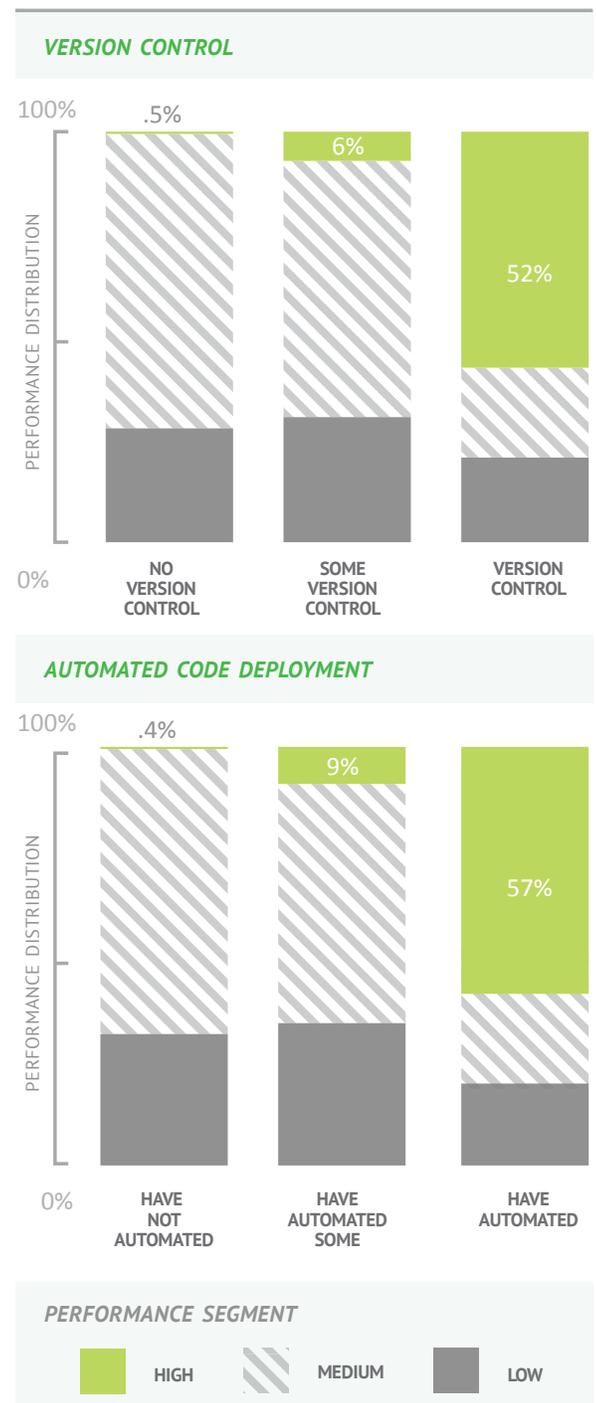
These practices have a direct impact on the key performance metrics that contribute to increased organizational agility and reliability.

Using version control as your single source of truth gives you the ability to pinpoint the cause of failures, easily roll back to a known state, and quickly deploy new service instances. Everything required to run an application, including the infrastructure and configuration code, should be in version control.

Automating code deployments provides several benefits that directly contribute to high performance. First, automating the configuration of your development, test, and production machines eliminates configuration drift between environments—a common point of failure in the deployment process. Second, automation significantly reduces your change lead time by replacing manual workflows with a consistent, repeatable process. Third, automation tools give visibility into the impact of changes before they're promoted to production, reducing risk.

Version control and automation together enable the highest levels of efficiency and productivity. Developers and QA engineers are able to request and provision properly configured environments in one command. This removes Operations as a bottleneck, empowers development/QA teams, and enables everyone to focus on things that matter to the business.

ORGANIZATIONS USING VERSION CONTROL AND AUTOMATED CODE DEPLOYMENT ARE HIGHER PERFORMING



Identifying Barriers to DevOps Adoption

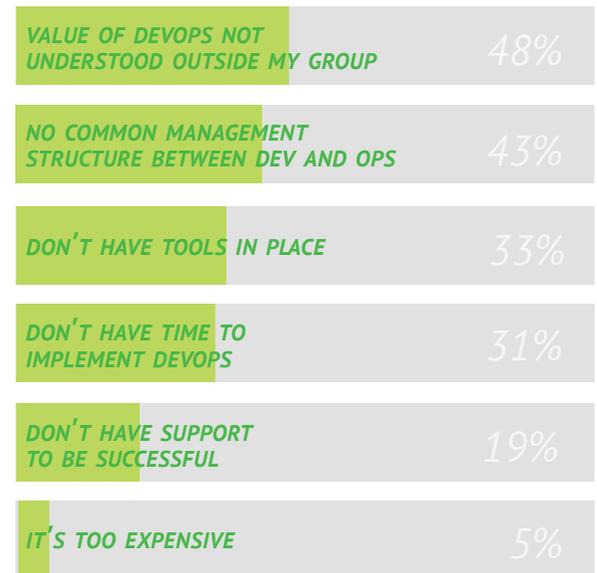
DevOps processes and tooling contribute to high performance, but these practices alone aren't enough to achieve organizational success. The most common barriers to DevOps adoption are cultural: lack of manager or team buy-in, or the value of DevOps isn't understood outside of a specific group. These barriers come from communication breakdowns, and working to solve them may help address other commonly identified blockers.

How common are these cultural barriers?

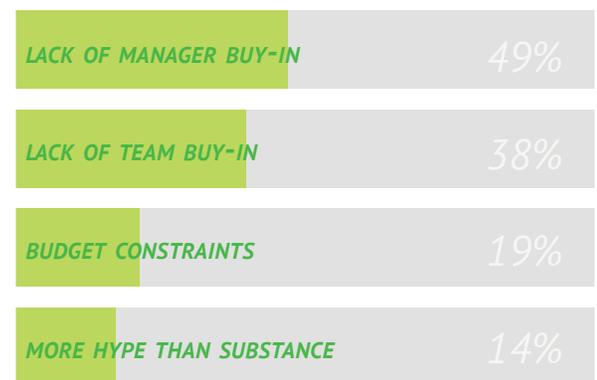
- 49 percent of respondents who had no plans to implement DevOps identified a lack of manager buy-in as a blocker—this was the most frequently identified barrier for this group, followed by "lack of team buy-in" (38%).
- 48 percent of all respondents indicated that one of the biggest difficulties in implementing DevOps was that the value wasn't understood outside of their group.

The best way to overcome these barriers is to start a conversation. Invite someone from another team to lunch, or ask them about their laptop sticker. Find out about the problems their team is facing, and tell them about your challenges. It's likely that there are skills and processes that can be shared between teams to solve specific problems, and they just haven't been identified yet. Creating these open channels of communication builds empathy and helps break down silos between Operations and Development teams.

BIGGEST DIFFICULTIES IN IMPLEMENTING DEVOPS



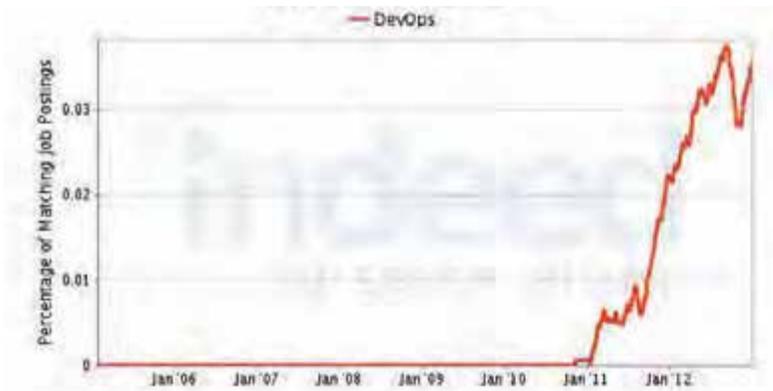
THOSE WHO HAD NO PLANS TO IMPLEMENT DEVOPS CITED THE FOLLOWING REASONS:



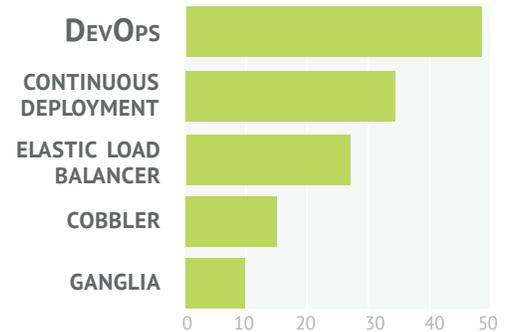
Demand for DevOps Skills is Growing

As DevOps adoption continues to accelerate, demand for DevOps skills has followed. Job listings for "DevOps" increased by 75 percent from January 2012 to January 2013 (Indeed.com), and mentions of "DevOps" as a skill increased by 50 percent (LinkedIn.com).

INDEED

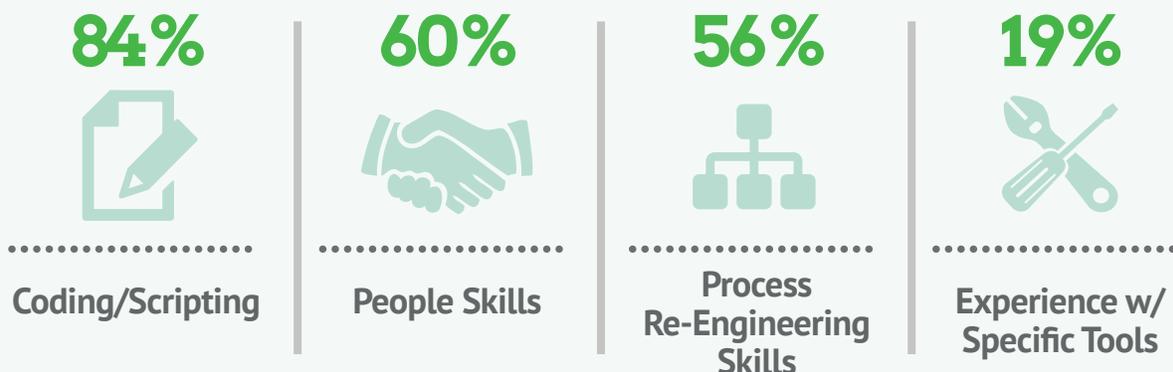


RELATIVE GROWTH: MENTIONS OF DEVOPS AS A SKILL



We wanted to know what constitutes the "DevOps skills" IT professionals need to stay ahead of the curve. We found coding and scripting high on the list of coveted skills, reflecting another emerging trend in IT: the need to automate manual tasks with modular, sharable bits of code. People skills were next, because communication and collaboration are the key to DevOps success. Process re-engineering was also popular, indicating a need for a holistic view of the system, rather than one-off solutions.

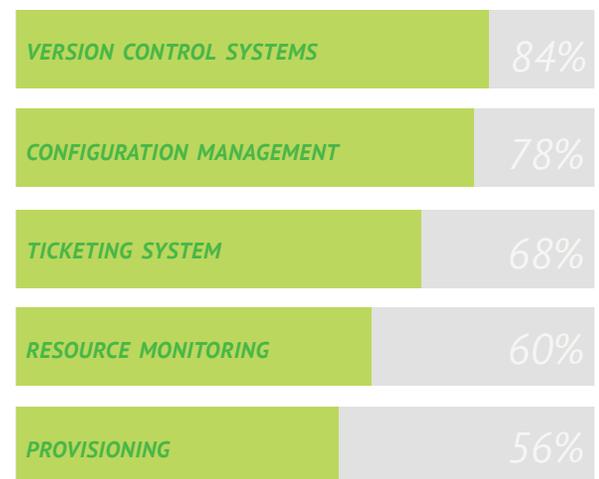
Desired skills for DevOps roles



Interestingly, experience with specific tools was not a priority. You can teach the tools more readily than you can teach the other skills. Regardless of the tools you choose, sharing the same toolchain eases communication across teams, allowing everyone to speak a common language. Version control systems and configuration management tools were seen as the biggest enablers of DevOps.

One way to start building a common toolchain is to talk to your Development or Operations team and see if there are ways you can apply their tools to what you do. Consider identifying a common pain point between both teams—such as deploying from dev to production—and see if you can jointly improve that process.

TOP 5 TOOLS USED TO SUPPORT DEVOPS INITIATIVES



Recommendations for Implementing DevOps

As DevOps adoption continues to accelerate and more organizations demonstrate success, those that aren't practicing DevOps are at risk of being left behind. To stay ahead of the curve, and achieve higher levels of performance, organizations need to foster those skills within their teams.

RECOMMENDATIONS FOR SYSADMINS, DEVELOPERS, OR SQA ENGINEERS

Automate. Automate. Automate.

Automation is the single biggest driver of high performance, increasing the overall quality and speed of code deployments. Greenfield environments have the advantage of not being bogged down by legacy processes and technical debt, but even established IT organizations can make incremental improvements using automation. Automate a single pain point such as DNS, NTP, or root passwords. Start small, prove the value, and use the visibility that success brings to tackle bigger projects.

Break down cultural barriers.

DevOps doesn't require buy-in from the whole company. If you're in Operations, find a developer who writes the code you deploy. If you're a developer, find one of the ops people who deploys your code. Have coffee. Hang out. Building relationships "across the aisle" will increase everyone's understanding of the problems facing different parts of the organization, which goes a long way towards getting everyone working towards the same goals.

Pick one source of truth and make it so.

Consolidate multiple sources of information into one source of truth by creating synchronization scripts for your HR system, CMDB, Asset DB, Policy DB, etc. Whether you use a service, a database (SQL or Hiera on disk), or pure data in version control (a YAML or JSON file), the important thing is that all data inputs to your configuration state are stored centrally and accessible via your configuration management system.

Learn the tools.

Sharing a common toolchain can help foster communication across teams and spread empathy about the challenges they face.

Foster DevOps skills within your team.

You almost certainly have people with DevOps skills already working for you. Support them. Listen to their ideas and help them succeed. You don't need to hire a DevOps team and create yet another functional silo. Instead, experiment with embedding ops and dev people on the opposite team, or creating a cross-functional team responsible for delivering a specific product or service.

Develop and use metrics.

Metrics are critical to tell the story of your success. They help you understand how you and your team are performing as well as help others understand why the DevOps investment is worthwhile. Use agility and reliability metrics such as deploy rate, change lead time, change failure rate, and mean time to recover to show business value. Use functional metrics like test cycle time, deployment time, defect rate in production and helpdesk ticket counts to demonstrate your success.

Organizations that follow these practices will not only increase agility and reliability, they will also have happier, more productive employees. Employees who know how to foster these environments will have more opportunities for growth as demand for these skills continues to grow. In the end, everyone wins—employees, the business, and your customers.

Encourage lateral communication.

Foster a culture of direct communication between peers, rather than using the top-down approach. Often, the best ideas will bubble up from the bottom: The more people that are collaborating, the more dynamic the exchange of ideas.

Key Findings

In December of 2012, Puppet Labs and IT Revolution Press surveyed over 4,000 IT Operations professionals and developers in the largest industry survey of this magnitude. The survey results revealed accelerating adoption of DevOps practices in IT organizations across companies of all sizes. Sixty-three percent of respondents have implemented DevOps practices, a 26 percent increase in adoption rate since 2011.¹

Respondents from organizations that implemented DevOps reported improved software deployment quality and more frequent software releases. Our findings substantiate the business value of DevOps, too: It enables high performance by increasing agility and reliability. We found that high performing organizations:

Ship code 30x faster

and complete those deployments 8,000 times faster than their peers.

Have 50% fewer failures

and restore service 12 times faster than their peers.

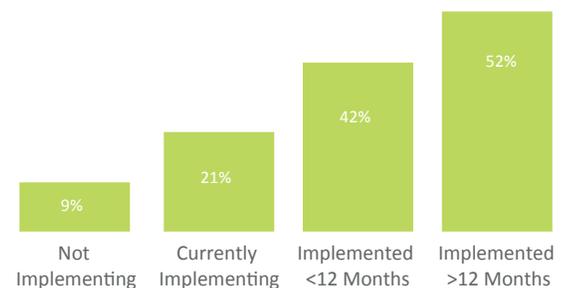
Not surprisingly, organizations that have implemented DevOps practices are up to five times more likely to be high-performing than those that have not. In fact, the longer organizations have been using DevOps practices, the better their performance: The best are getting better.

DevOps adoption is at a tipping point. Adopting DevOps practices improves performance, and the longer those practices are in place, the greater the performance advantage. Waiting to implement is no longer an option. Following our recommendations will put you ahead of the curve as DevOps approaches critical mass.

ORGANIZATIONS THAT HAVE IMPLEMENTED DEVOPS SAW THESE BENEFITS:



HIGH PERFORMANCE BY DEVOPS MATURITY



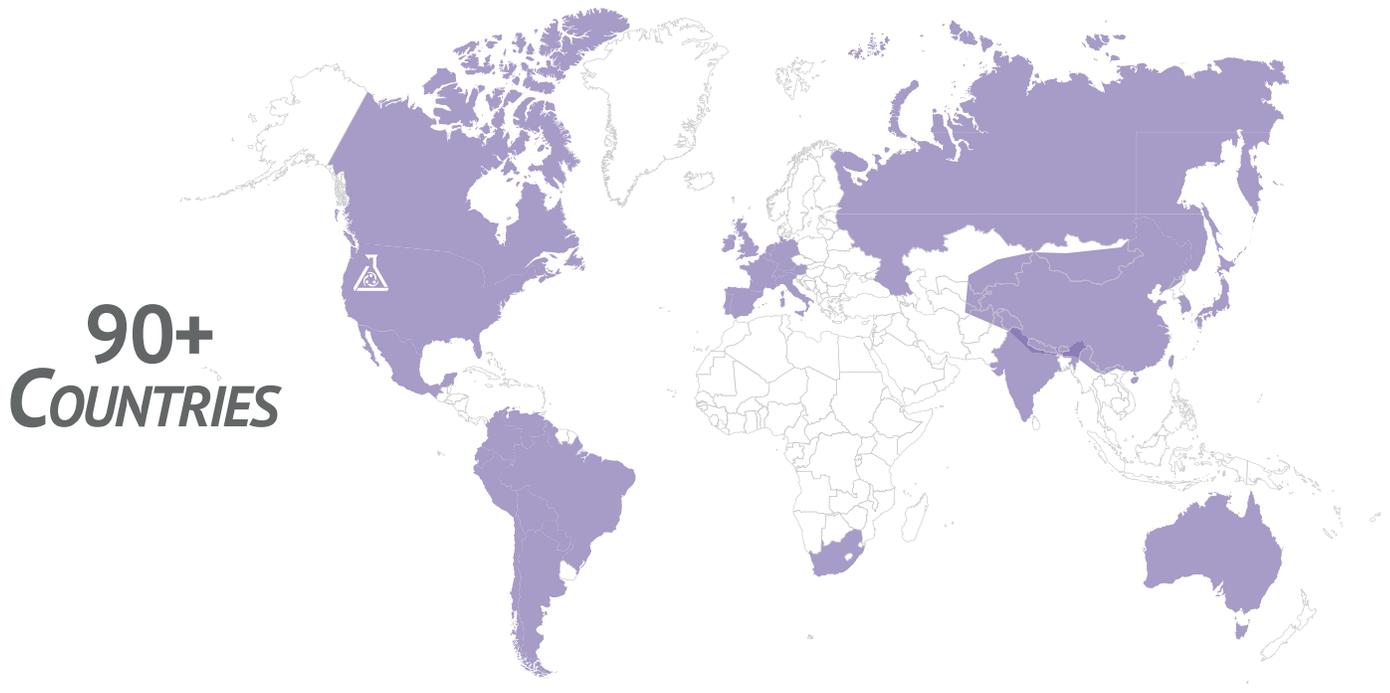
¹ Respondents of both the 2011 and 2012 survey are likely to be early adopters, making the increase in DevOps adoption rate much more telling of the overall growth of DevOps during this time period.

Who Took The Survey?

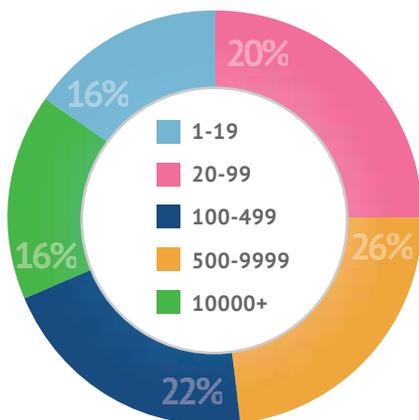
We saw respondents from over 90 countries across organizations of all sizes. The majority of respondents identified themselves as administrators, engineers, or developers, with only 16 percent manager-level or above.

Respondents worked for organizations running the gamut from startups to small/medium enterprises to WebOps giants, revealing that DevOps is happening everywhere.

SURVEY AREA



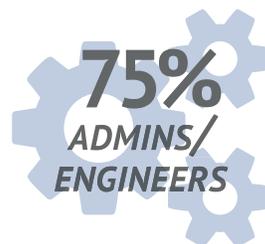
COMPANY SIZE (# OF EMPLOYEES)



DEPARTMENTS



ROLES



DevOps Increases Agility and Reliability

IT organizations are expected to respond more quickly to urgent business needs while simultaneously providing stable, secure, and predictable IT service. However, the systems on which the business operates are typically fragile and hostile to change. Adopting Agile development processes without improving operational reliability or communication between developers and operations only makes this problem worse. The increased frequency of releases from development creates even more of a burden on an already strained IT organization. Similarly, adopting rigid ITIL/ITSM standards without addressing development issues and improving communication channels results in an inflexible IT organization that simply cannot respond to business needs quickly enough. DevOps picks up where Agile methodology and IT standards left off.

For the first time ever, by adopting DevOps practices, IT organizations can be simultaneously agile and reliable, thus fulfilling the promise of over 20 years of Agile and ITIL/ITSM.

High performing organizations:

Deploy code 30 times more frequently.

High performing organizations deploy code 30 times more often, and 8000 times faster than their peers, deploying multiple times a day, versus an average of once a month. Frequent deployments coupled with faster change lead times enable operational agility.

Have 50 percent fewer failures.

High performing organizations have double the change success rate and restore service 12 times faster than their peers. Fewer failures and faster recovery mean less risk to the business when changes are deployed.

DevOps practices enable IT organizations to quickly and safely deploy changes, freeing them to work on higher level business objectives. In the next section, we break out four key metrics that show the impact DevOps has on performance.



Rob Cornish, CTO at the International Securities Exchange (ISE), a leading US options exchange that exceeds daily trading volumes of 2.5 million contracts.

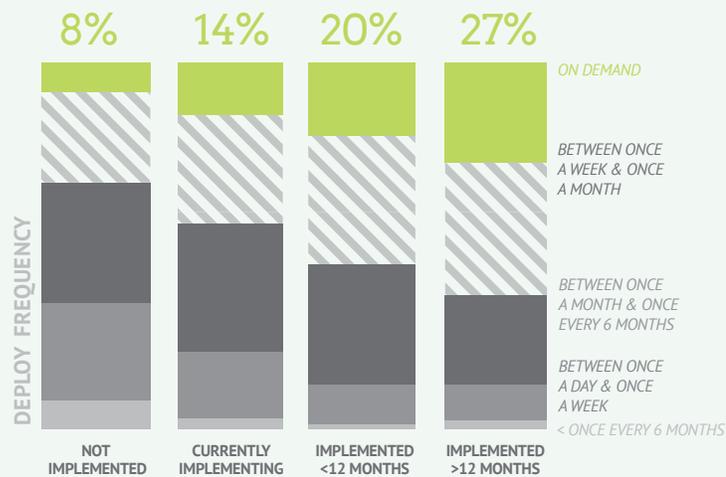
"DevOps has been instrumental in reducing our software deployment cycle from days to minutes. We are able to deploy new features to our customers quickly and safely, and rapidly respond to market changes."

How DevOps Enables High Performance

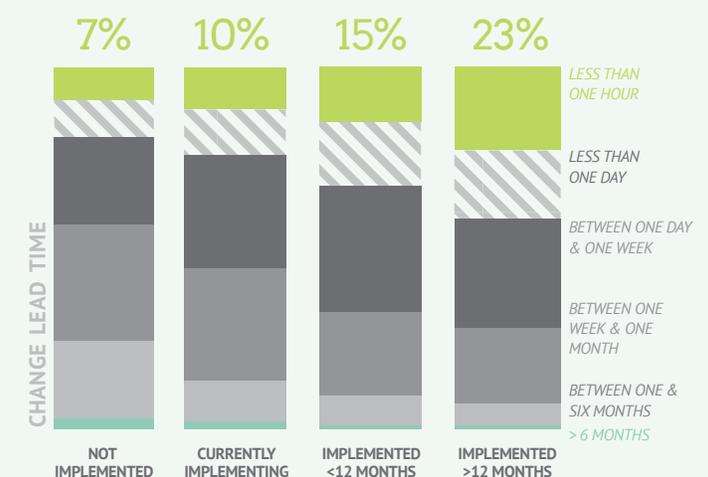
A high-performing organization is characterized by its ability to ship business-critical applications quickly without disrupting service. The classic examples of this are big WebOps shops—Google, Amazon, Twitter, and Etsy—known for deploying multiple times a day.

We found that organizations that had implemented DevOps practices were up to five times more likely to be high-performing than those that had not, and the more mature the implementation, the higher the performance.

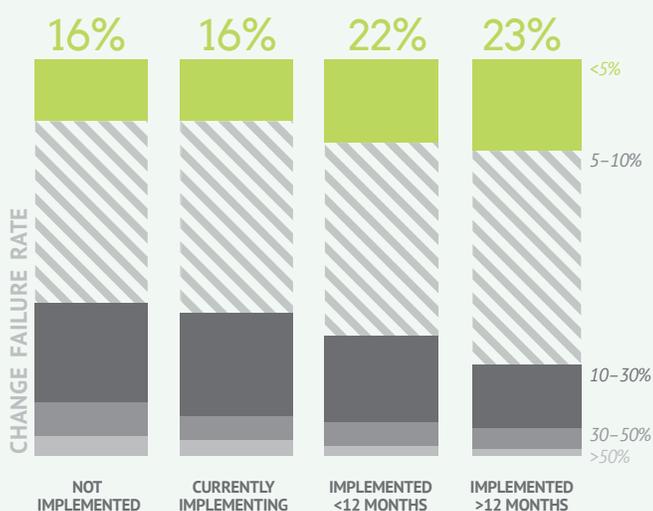
DEPLOY FREQUENCY



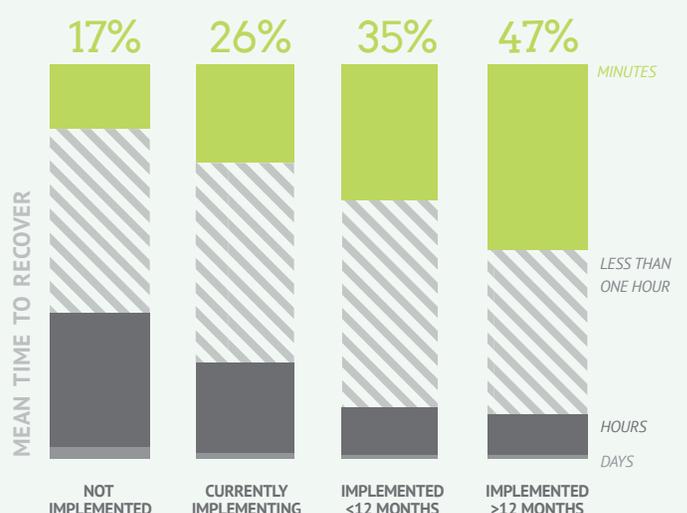
CHANGE LEAD TIME



CHANGE FAILURE RATE



MEAN TIME TO RECOVER



We analyzed four key DevOps performance metrics—deploy frequency, change lead time, change failure rate, and mean time to recover—by DevOps maturity, ranging from not implemented to implemented over 12 months ago. Organizations with more mature DevOps implementations saw significantly higher performance across all metrics compared to those that had not yet implemented DevOps.

Key Agility Performance Indicators

DEPLOYMENT FREQUENCY

High-performing organizations deploy at least once a week, and often multiple times a day. On average, this is 95 percent less time between deployments than lower performing organizations, enabling these organizations to quickly respond to market changes or customer feedback, and iterate on new ideas.

CHANGE LEAD TIME

Being able to quickly make changes is a basic measure of agility. If your infrastructure is bogged down by technical debt, the time to get code successfully running in production is significantly longer and your changes are more likely to fail. High-performing organizations make changes with a few minutes' notice, while their peers have change lead times of up to a month. Agile organizations can make 8,000 changes before their slower competitors can vet and deploy a single change.

Key Reliability Performance Indicators

MEAN TIME TO RECOVER

High-performing organizations recover from outages within minutes, compared to lower performing organizations that take an hour or longer to recover: over 30 times faster. Most organizations can't afford even a few minutes of downtime—that can mean thousands of lost transactions, resulting in lost revenue and lower customer satisfaction.

CHANGE FAIL RATE

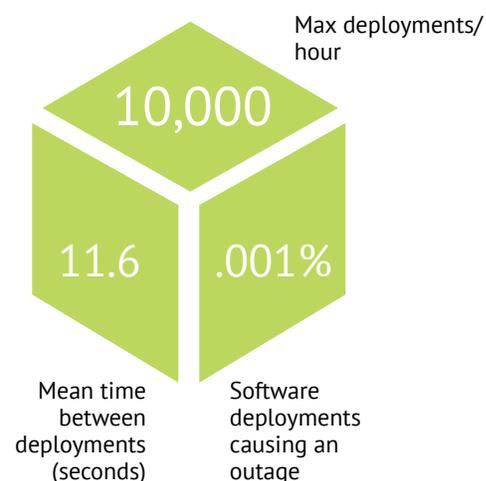
When changes fail, your organization has to devote time and resources to recovering, usually by taking those resources from developing, releasing, and supporting new products or updates. High-performing organizations have over 50 percent fewer failures from code changes. Changes often fail because the development and test environments don't match the production environment.

Obamas DevOps Success



Using DevOps practices, Obama's tech team was able to manage 2,000 nodes across three data centers, processing over 180TB and 8.5 billion requests, with just 30 minutes of downtime during the entire 18 months of the election campaign.²

DEVOPS VALUE IN ACTION: VELOCITY AT AMAZON AWS³



²Condliffe, Jamie, "Inside Obama's Tech Team." <http://gizmodo.com/5960790/inside-obamas-team-tech>. Nov 15, 2012.

³Jenkins, Jon. "Velocity Culture." Keynote, Velocity Conference, Santa Clara, Jun 16, 2011.

How to Achieve High Performance

Now that we know what high performance looks like, how do you actually achieve it? High-performing organizations share two common practices:

- 89 percent use version control systems for infrastructure management
- 82 percent have automated their code deployments

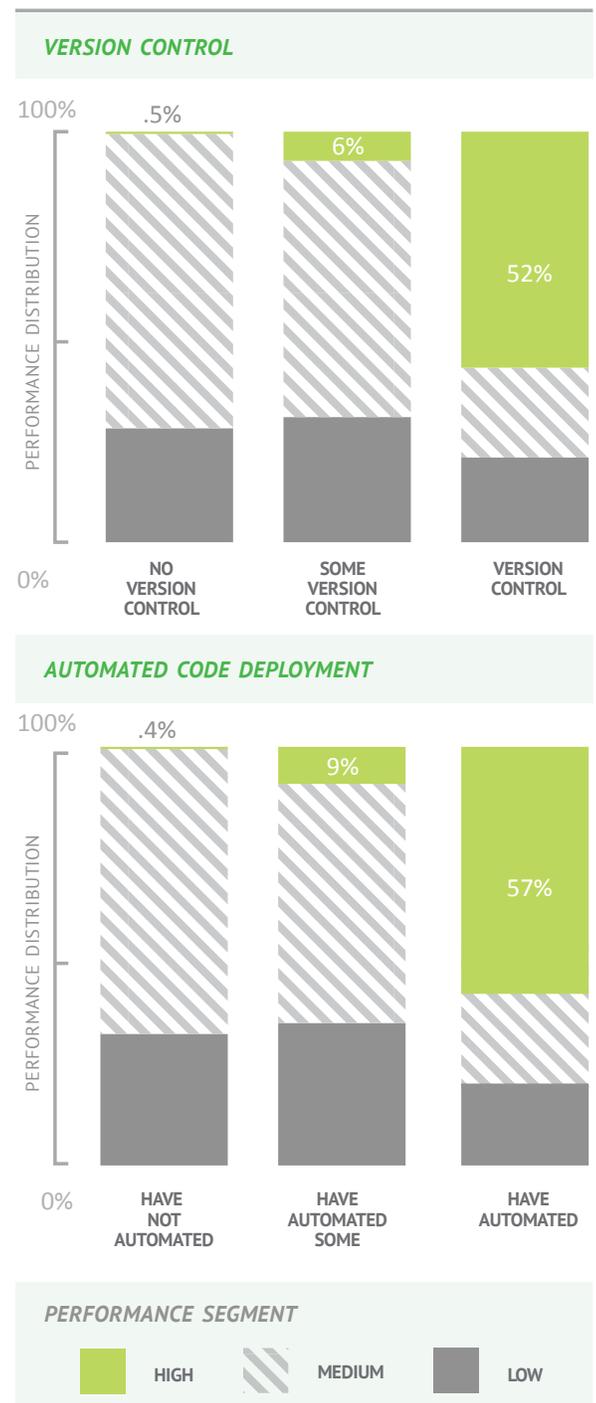
These practices have a direct impact on the key performance metrics that contribute to increased organizational agility and reliability.

Using version control as your single source of truth gives you the ability to pinpoint the cause of failures, easily roll back to a known state, and quickly deploy new service instances. Everything required to run an application, including the infrastructure and configuration code, should be in version control.

Automating code deployments provides several benefits that directly contribute to high performance. First, automating the configuration of your development, test, and production machines eliminates configuration drift between environments—a common point of failure in the deployment process. Second, automation significantly reduces your change lead time by replacing manual workflows with a consistent, repeatable process. Third, automation tools give visibility into the impact of changes before they're promoted to production, reducing risk.

Version control and automation together enable the highest levels of efficiency and productivity. Developers and QA engineers are able to request and provision properly configured environments in one command. This removes Operations as a bottleneck, empowers development/QA teams, and enables everyone to focus on things that matter to the business.

ORGANIZATIONS USING VERSION CONTROL AND AUTOMATED CODE DEPLOYMENT ARE HIGHER PERFORMING



Identifying Barriers to DevOps Adoption

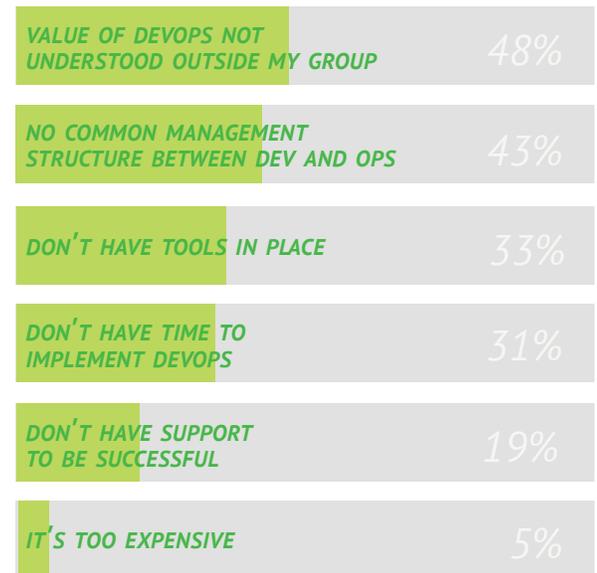
DevOps processes and tooling contribute to high performance, but these practices alone aren't enough to achieve organizational success. The most common barriers to DevOps adoption are cultural: lack of manager or team buy-in, or the value of DevOps isn't understood outside of a specific group. These barriers come from communication breakdowns, and working to solve them may help address other commonly identified blockers.

How common are these cultural barriers?

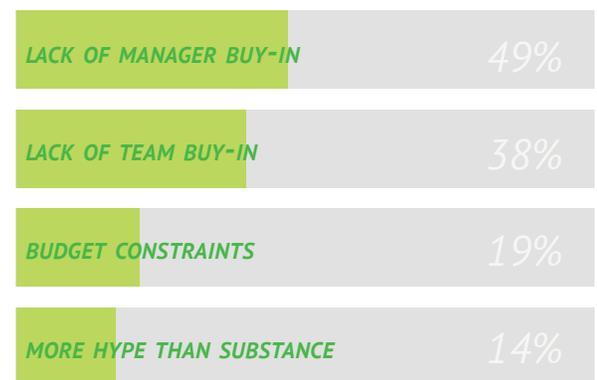
- 49 percent of respondents who had no plans to implement DevOps identified a lack of manager buy-in as a blocker—this was the most frequently identified barrier for this group, followed by "lack of team buy-in" (38%).
- 48 percent of all respondents indicated that one of the biggest difficulties in implementing DevOps was that the value wasn't understood outside of their group.

The best way to overcome these barriers is to start a conversation. Invite someone from another team to lunch, or ask them about their laptop sticker. Find out about the problems their team is facing, and tell them about your challenges. It's likely that there are skills and processes that can be shared between teams to solve specific problems, and they just haven't been identified yet. Creating these open channels of communication builds empathy and helps break down silos between Operations and Development teams.

BIGGEST DIFFICULTIES IN IMPLEMENTING DEVOPS



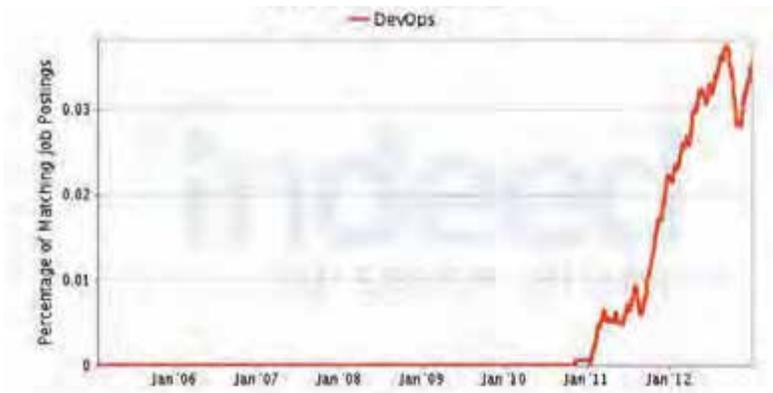
THOSE WHO HAD NO PLANS TO IMPLEMENT DEVOPS CITED THE FOLLOWING REASONS:



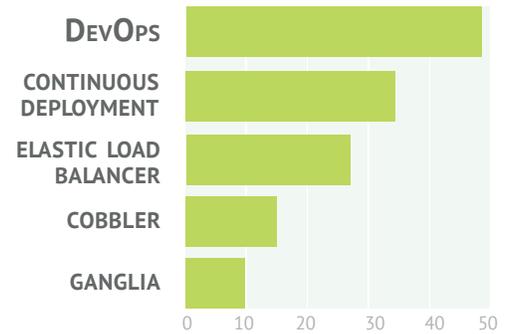
Demand for DevOps Skills is Growing

As DevOps adoption continues to accelerate, demand for DevOps skills has followed. Job listings for "DevOps" increased by 75 percent from January 2012 to January 2013 (Indeed.com), and mentions of "DevOps" as a skill increased by 50 percent (LinkedIn.com).

INDEED

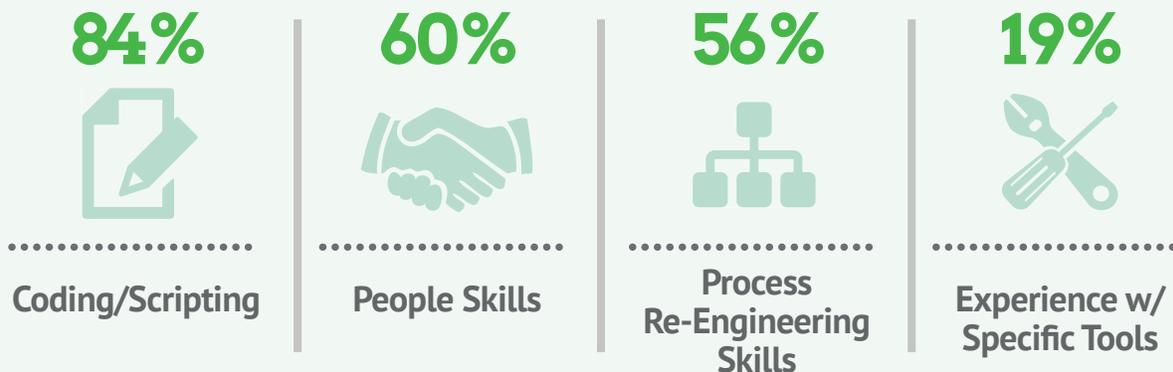


RELATIVE GROWTH: MENTIONS OF DEVOPS AS A SKILL



We wanted to know what constitutes the "DevOps skills" IT professionals need to stay ahead of the curve. We found coding and scripting high on the list of coveted skills, reflecting another emerging trend in IT: the need to automate manual tasks with modular, sharable bits of code. People skills were next, because communication and collaboration are the key to DevOps success. Process re-engineering was also popular, indicating a need for a holistic view of the system, rather than one-off solutions.

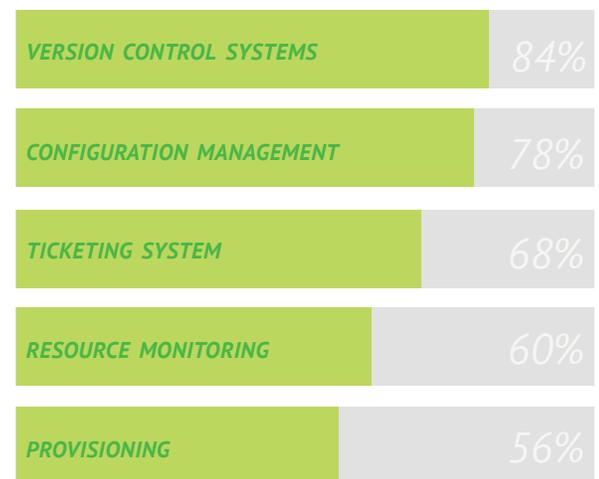
Desired skills for DevOps roles



Interestingly, experience with specific tools was not a priority. You can teach the tools more readily than you can teach the other skills. Regardless of the tools you choose, sharing the same toolchain eases communication across teams, allowing everyone to speak a common language. Version control systems and configuration management tools were seen as the biggest enablers of DevOps.

One way to start building a common toolchain is to talk to your Development or Operations team and see if there are ways you can apply their tools to what you do. Consider identifying a common pain point between both teams—such as deploying from dev to production—and see if you can jointly improve that process.

TOP 5 TOOLS USED TO SUPPORT DEVOPS INITIATIVES



Recommendations for Implementing DevOps

As DevOps adoption continues to accelerate and more organizations demonstrate success, those that aren't practicing DevOps are at risk of being left behind. To stay ahead of the curve, and achieve higher levels of performance, organizations need to foster those skills within their teams.

RECOMMENDATIONS FOR SYSADMINS, DEVELOPERS, OR SQA ENGINEERS

Automate. Automate. Automate.

Automation is the single biggest driver of high performance, increasing the overall quality and speed of code deployments. Greenfield environments have the advantage of not being bogged down by legacy processes and technical debt, but even established IT organizations can make incremental improvements using automation. Automate a single pain point such as DNS, NTP, or root passwords. Start small, prove the value, and use the visibility that success brings to tackle bigger projects.

Break down cultural barriers.

DevOps doesn't require buy-in from the whole company. If you're in Operations, find a developer who writes the code you deploy. If you're a developer, find one of the ops people who deploys your code. Have coffee. Hang out. Building relationships "across the aisle" will increase everyone's understanding of the problems facing different parts of the organization, which goes a long way towards getting everyone working towards the same goals.

Pick one source of truth and make it so.

Consolidate multiple sources of information into one source of truth by creating synchronization scripts for your HR system, CMDB, Asset DB, Policy DB, etc. Whether you use a service, a database (SQL or Hiera on disk), or pure data in version control (a YAML or JSON file), the important thing is that all data inputs to your configuration state are stored centrally and accessible via your configuration management system.

Learn the tools.

Sharing a common toolchain can help foster communication across teams and spread empathy about the challenges they face.

Foster DevOps skills within your team.

You almost certainly have people with DevOps skills already working for you. Support them. Listen to their ideas and help them succeed. You don't need to hire a DevOps team and create yet another functional silo. Instead, experiment with embedding ops and dev people on the opposite team, or creating a cross-functional team responsible for delivering a specific product or service.

Develop and use metrics.

Metrics are critical to tell the story of your success. They help you understand how you and your team are performing as well as help others understand why the DevOps investment is worthwhile. Use agility and reliability metrics such as deploy rate, change lead time, change failure rate, and mean time to recover to show business value. Use functional metrics like test cycle time, deployment time, defect rate in production and helpdesk ticket counts to demonstrate your success.

Organizations that follow these practices will not only increase agility and reliability, they will also have happier, more productive employees. Employees who know how to foster these environments will have more opportunities for growth as demand for these skills continues to grow. In the end, everyone wins—employees, the business, and your customers.

Encourage lateral communication.

Foster a culture of direct communication between peers, rather than using the top-down approach. Often, the best ideas will bubble up from the bottom: The more people that are collaborating, the more dynamic the exchange of ideas.